

Conversions between Taylor Series and Orthogonal Polynomials

Marc A. Murison

Astronomical Applications Department
U.S. Naval Observatory
Washington, DC

murison@aa.usno.navy.mil
<http://aa.usno.navy.mil/murison/>

May 14, 1999

[-] Introduction

This worksheet illustrates my **polynomials** Maple package. Its purpose is to allow one to convert a truncated Taylor series polynomial in some variable x ,

$$S_N(x) = \sum_{k=0}^N a_k x^k$$

to an equivalent orthogonal polynomial,

$$S_N(x) = \sum_{k=0}^N c_k Q_k(x)$$

where Q is one of P, T, U, H, L, or G. These stand for the following orthogonal polynomials:

P	Legendre
T	Chebyshev type 1
U	Chebyshev type 2
H	Hermite
L	Laguerre
G	Gegenbauer (ultraspherical)

There is also a conversion procedure to transform back to a truncated Taylor series. Use of the conversion procedures is documented by way of examples in the next section. Following that section is a numerical illustration. Finally, the last section contains the source code of the **polynomials** package.

[-] Usage Examples

- Setup

```
restart
with(polynomials)

[convert/G, convert/H, convert/L, convert/P, convert/T, convert/U, convert/taylor, init, orthosubs,
plotpoly]

s := series( cos(x + φ) sinh(x) e(-x)y + x tan(x), x, 4)
s := cos(φ) x + (-cos(φ) y - sin(φ) + 1) x2 +  $\left(\frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y\right) x^3 + O(x^4)$ 
```

- A convenient plotting procedure

```
plotpoly := proc(intlist::list(posint), ptype::name)
local p, k, xmin, xmax, ymin, ymax, ytext, ttext, pargs;
if ptype = 'G' then pargs := args[3], x else pargs := x fi;
xmin := -1;
xmax := 1;
ymin := -1;
ymax := 1;
if ptype = 'L' then xmin := 0; xmax := 5; ymin := -5; ymax := 5
elif ptype = 'H' then xmin := 0; xmax := 3; ymin := -2; ymax := 8
elif ptype = 'U' then ymin := -5; ymax := 5
elif ptype = 'G' and 1 / 2 < args[3] then ymin := -5; ymax := 5
fi;
p := [ ];
for k to nops(intlist) do
if ptype = 'H' then p := [ op(p), plot(orthopoly[ptype](intlist[k], pargs) / (intlist[k]^3),
x = xmin .. xmax, color = blue) ]
else p := [ op(p), plot(orthopoly[ptype](intlist[k], pargs), x = xmin .. xmax, color = blue) ]
fi
od;
ytext := cat(ptype, "[n](x)");
if ptype = 'P' then ttext := "Legendre"
elif ptype = 'T' then ttext := "Chebyshev type 1"
elif ptype = 'U' then ttext := "Chebyshev type 2"
elif ptype = 'H' then ttext := "Hermite"; ytext := cat(ptype, "[n](x)/n^3")
elif ptype = 'L' then ttext := "Laguerre"
elif ptype = 'G' then ttext := cat("Gegenbauer (a=", convert(args[3], string), ")")
fi;
plots[display]([ op(p), plot(0, x = xmin .. xmax, color = black)],
view = [ xmin .. xmax, ymin .. ymax], title = ttext, labels = [ "x", ytext ], axes = box)
end
```

- Legendre

```

convert(s, P, x)


$$\left( \frac{3}{10} \cos(\phi) y^2 + \frac{4}{5} \cos(\phi) + \frac{3}{5} \sin(\phi) y \right) P_1(x) + \left( -\frac{2}{3} \cos(\phi) y - \frac{2}{3} \sin(\phi) + \frac{2}{3} \right) P_2(x)$$


$$+ \left( \frac{1}{5} \cos(\phi) y^2 - \frac{2}{15} \cos(\phi) + \frac{2}{5} \sin(\phi) y \right) P_3(x) - \frac{1}{3} \cos(\phi) y - \frac{1}{3} \sin(\phi) + \frac{1}{3}$$


convert(%, taylor, x)


$$\cos(\phi) x + (-\cos(\phi) y - \sin(\phi) + 1) x^2 + \left( \frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y \right) x^3$$

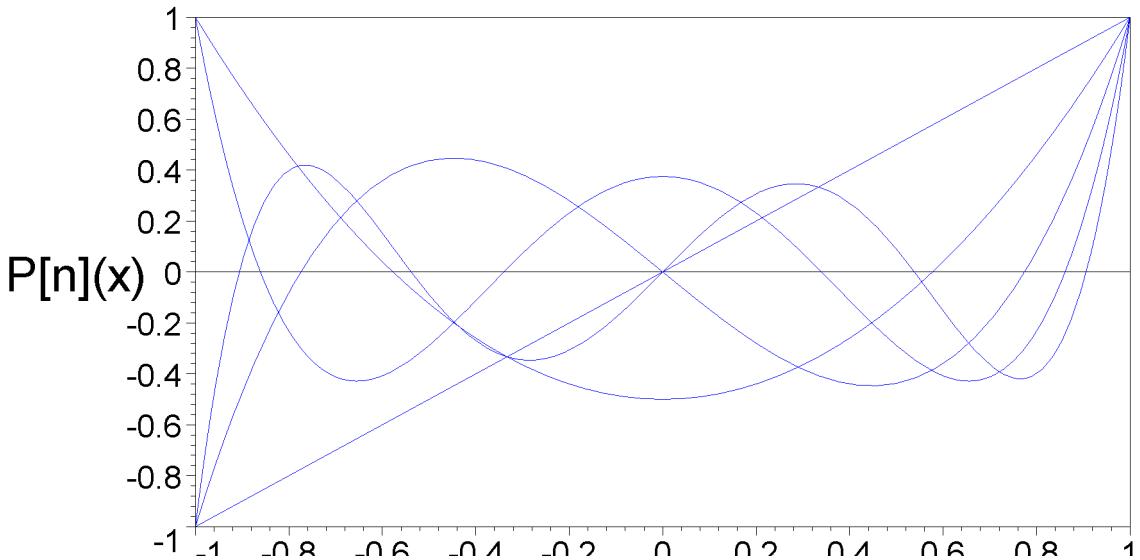

simplify(convert(s, polynom) - %)

0

plotpoly([1, 2, 3, 4, 5], P)

```

Legendre



- Chebyshev type 1

```

convert(s, T, x)


$$\left( \frac{3}{8} \cos(\phi) y^2 + \frac{3}{4} \cos(\phi) + \frac{3}{4} \sin(\phi) y \right) T_1(x) + \left( -\frac{1}{2} \cos(\phi) y - \frac{1}{2} \sin(\phi) + \frac{1}{2} \right) T_2(x)$$


$$+ \left( \frac{1}{8} \cos(\phi) y^2 - \frac{1}{12} \cos(\phi) + \frac{1}{4} \sin(\phi) y \right) T_3(x) - \frac{1}{2} \cos(\phi) y - \frac{1}{2} \sin(\phi) + \frac{1}{2}$$


convert(%, taylor, x)


$$\cos(\phi) x + (-\cos(\phi) y - \sin(\phi) + 1) x^2 + \left( \frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y \right) x^3$$

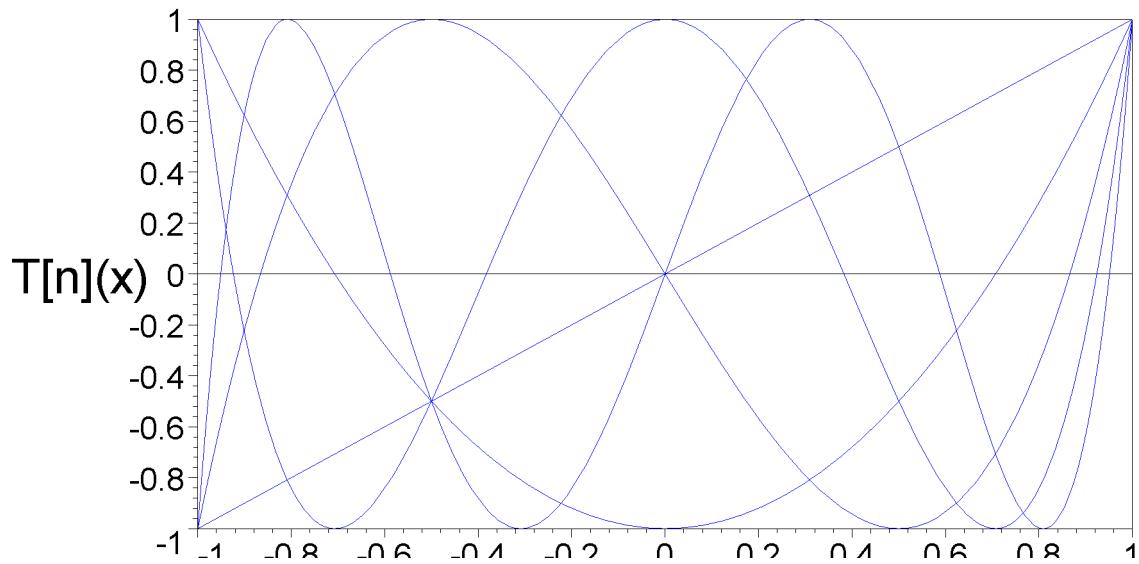

```

```
simplify(convert(s, polynom) - %)
```

```
0
```

```
plotpoly([ 1, 2, 3, 4, 5 ], T)
```

Chebyshev type 1



- Chebyshev type 2

```
convert(s, U, x)
```

$$\begin{aligned} & \left(\frac{1}{8} \cos(\phi) y^2 + \frac{5}{12} \cos(\phi) + \frac{1}{4} \sin(\phi) y \right) U_1(x) + \left(-\frac{1}{4} \cos(\phi) y - \frac{1}{4} \sin(\phi) + \frac{1}{4} \right) U_2(x) \\ & + \left(\frac{1}{16} \cos(\phi) y^2 - \frac{1}{24} \cos(\phi) + \frac{1}{8} \sin(\phi) y \right) U_3(x) - \frac{1}{4} \cos(\phi) y - \frac{1}{4} \sin(\phi) + \frac{1}{4} \end{aligned}$$

```
convert(%, taylor, x)
```

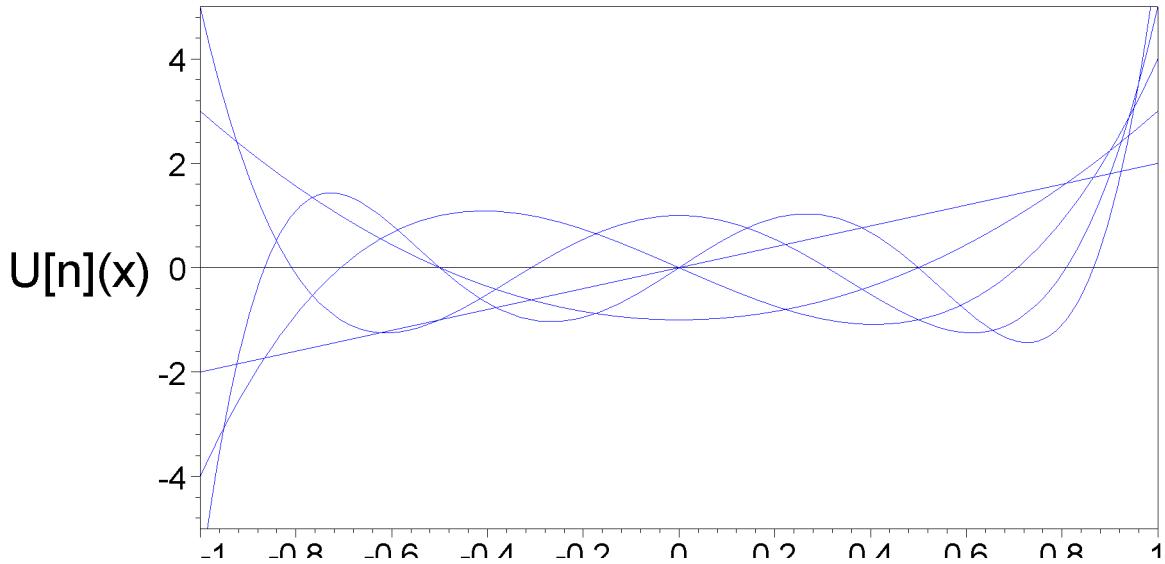
$$\cos(\phi) x + (-\cos(\phi) y - \sin(\phi) + 1) x^2 + \left(\frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y \right) x^3$$

```
simplify(convert(s, polynom) - %)
```

```
0
```

```
plotpoly([ 1, 2, 3, 4, 5 ], U)
```

Chebyshev type 2



- Hermite

```
convert(s, H, x)

$$\left( \frac{3}{8} \cos(\phi) y^2 + \frac{1}{4} \cos(\phi) + \frac{3}{4} \sin(\phi) y \right) H_1(x) + \left( -\frac{1}{4} \cos(\phi) y - \frac{1}{4} \sin(\phi) + \frac{1}{4} \right) H_2(x)$$


$$+ \left( \frac{1}{16} \cos(\phi) y^2 - \frac{1}{24} \cos(\phi) + \frac{1}{8} \sin(\phi) y \right) H_3(x) - \frac{1}{2} \cos(\phi) y - \frac{1}{2} \sin(\phi) + \frac{1}{2}$$

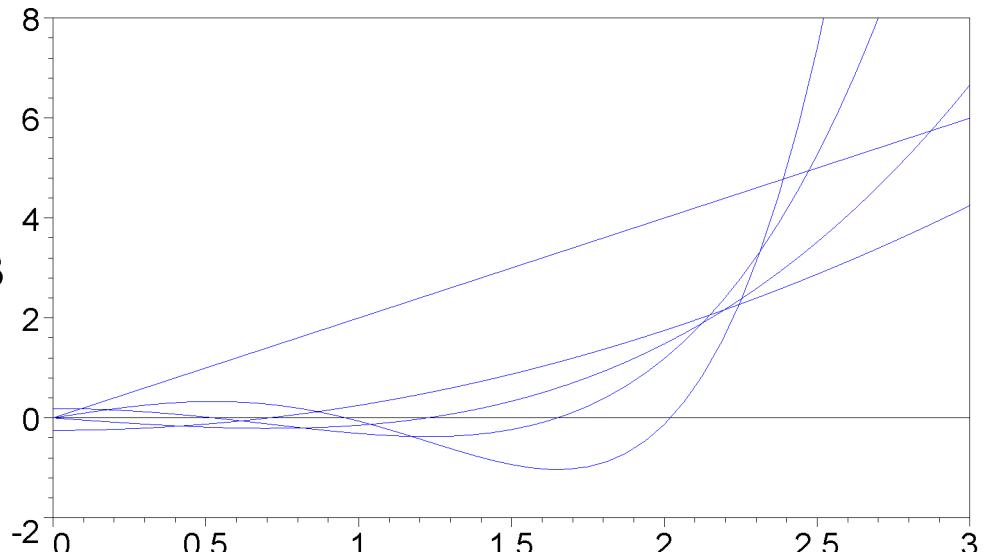
convert(% , taylor, x)

$$\cos(\phi) x + (-\cos(\phi) y - \sin(\phi) + 1) x^2 + \left( \frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y \right) x^3$$

simplify(convert(s, polynom) - %)
0
plotpoly([1, 2, 3, 4, 5], H)
```

Hermite

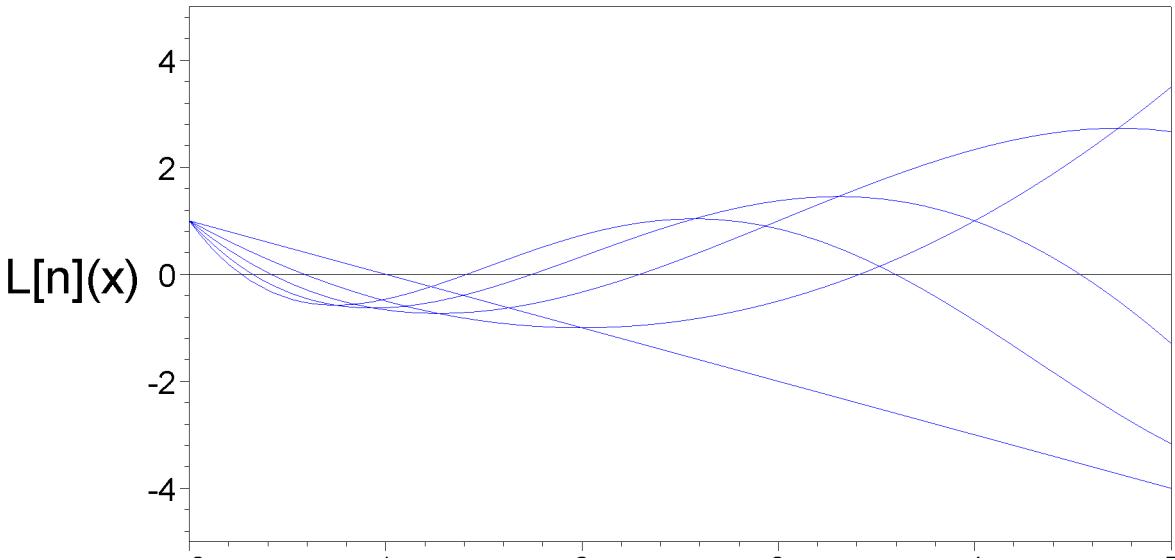
$H[n](x)/n^3$



- Laguerre

```
convert(s, L, x)
(-9 cos(phi) y2 + 5 cos(phi) - 18 sin(phi) y + 4 cos(phi) y + 4 sin(phi) - 4) L1(x)
+ (9 cos(phi) y2 - 6 cos(phi) + 18 sin(phi) y - 2 cos(phi) y - 2 sin(phi) + 2) L2(x)
+ (-3 cos(phi) y2 + 2 cos(phi) - 6 sin(phi) y) L3(x) + 3 cos(phi) y2 - cos(phi) + 6 sin(phi) y
- 2 cos(phi) y - 2 sin(phi) + 2
convert(%, taylor, x)
cos(phi) x + (-cos(phi) y - sin(phi) + 1) x2 +  $\left(\frac{1}{2} \cos(\phi) y^2 - \frac{1}{3} \cos(\phi) + \sin(\phi) y\right) x^3$ 
simplify(convert(s, polynom) - %)
0
plotpoly([1, 2, 3, 4, 5], L)
```

Laguerre



- Gegenbauer (ultraspherical)

convert(s, G, x, a)

$$\begin{aligned} & \left(\frac{1}{24} \frac{(3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y)(1 + 2a)}{a(1+a)(2+a)} \right. \\ & \quad \left. + \frac{1}{2} \frac{\frac{1}{12} \frac{3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y}{1+a} + \cos(\phi)}{a} \right) G_1(x) \\ & - \frac{1}{2} \frac{(\cos(\phi) y + \sin(\phi) - 1) G_2(x)}{a(1+a)} - \frac{1}{2} \frac{\cos(\phi) y + \sin(\phi) - 1}{1+a} \\ & + \frac{1}{8} \frac{(3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y) G_3(x)}{a(1+a)(2+a)} \end{aligned}$$

convert($\%, taylor, x, a$)

$$\begin{aligned} & \frac{1}{8} \frac{(3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y) \left(4a^2 + \frac{4}{3}a^3 + \frac{8}{3}a \right) x^3}{a(1+a)(2+a)} \\ & - \frac{1}{2} \frac{(\cos(\phi) y + \sin(\phi) - 1)(2a + 2a^2)x^2}{a(1+a)} + \left(2 \left(\right. \right. \end{aligned}$$

$$\left. \begin{aligned} & \frac{1}{24} \frac{(3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y)(1 + 2 a)}{a(1 + a)(2 + a)} \\ & + \frac{1}{2} \frac{\frac{1}{12} \frac{3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y}{1 + a} + \cos(\phi)}{a} \\ & + \frac{1}{8} \frac{(3 \cos(\phi) y^2 - 2 \cos(\phi) + 6 \sin(\phi) y)(-2 a^2 - 2 a)}{a(1 + a)(2 + a)} \end{aligned} \right\} x$$

simplify(convert(*s*, polynom) - %)

0

Note:

$$G_n(x, 0) = \frac{2 T_n(x)}{n},$$

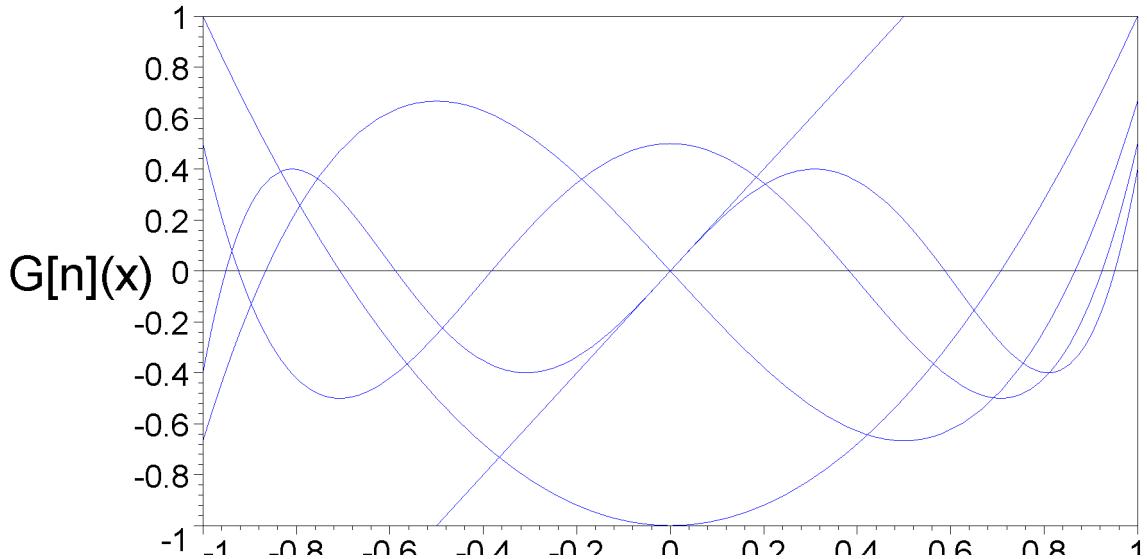
$$G_n(x, 1) = U_n(x),$$

and

$$G_n\left(x, \frac{1}{2}\right) = P_n(x).$$

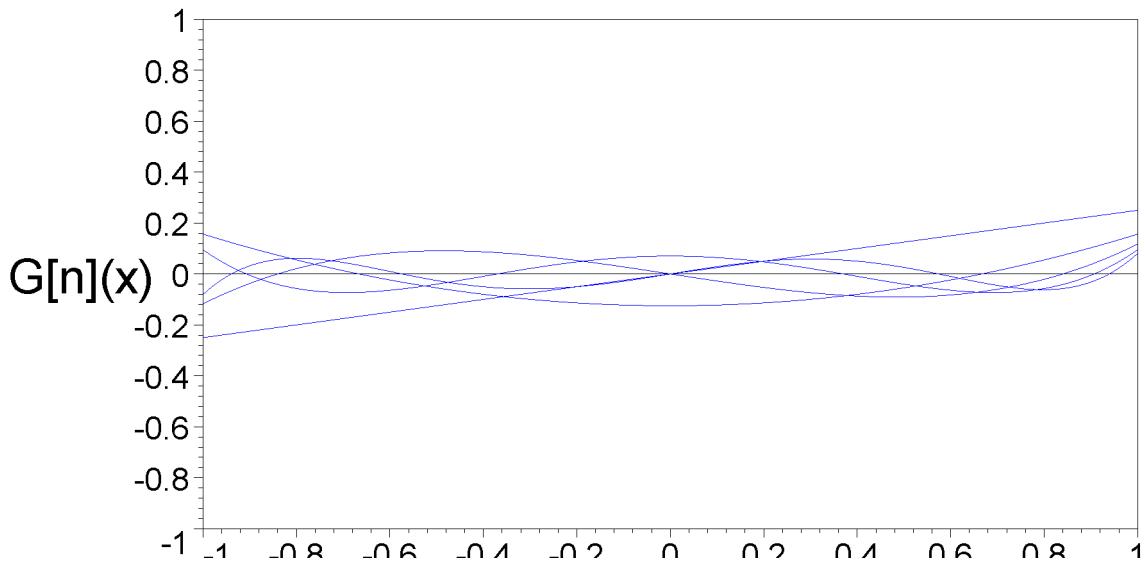
plotpoly([1, 2, 3, 4, 5], G, 0)

Gegenbauer (a=0)



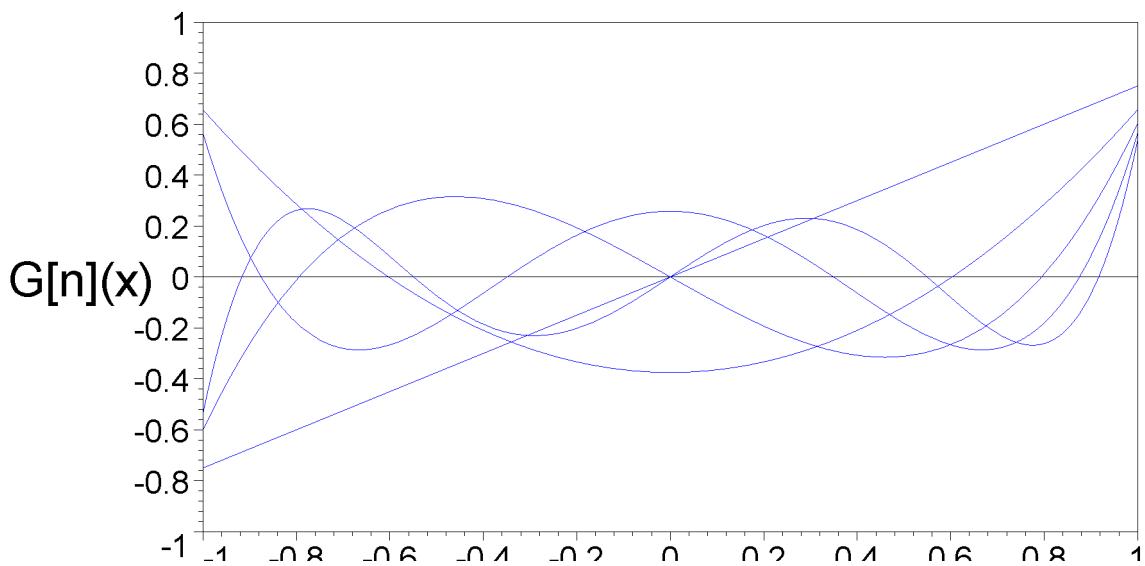
plotpoly([1, 2, 3, 4, 5], G, $\frac{1}{8}$)

Gegenbauer ($a=1/8$)



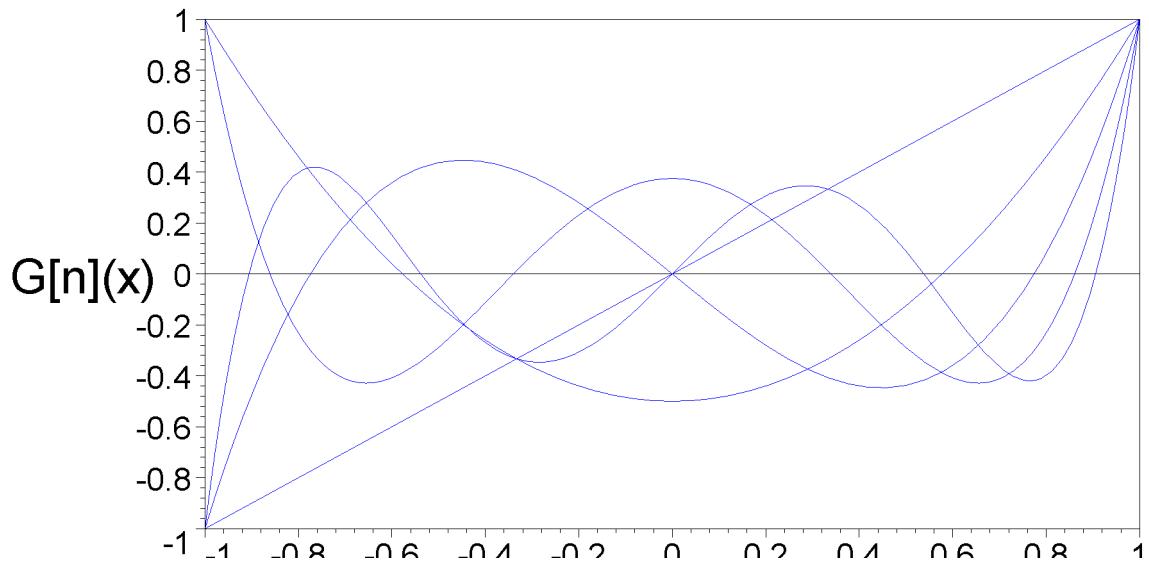
plotpoly $\left([1, 2, 3, 4, 5], G, \frac{3}{8}\right)$

Gegenbauer ($a=3/8$)



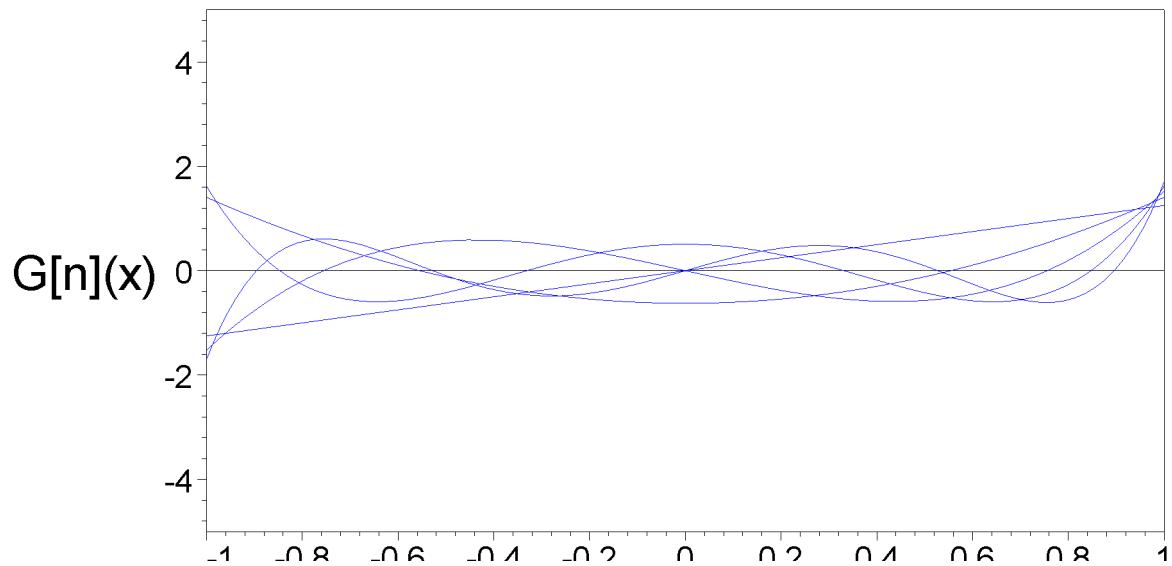
plotpoly $\left([1, 2, 3, 4, 5], G, \frac{1}{2}\right)$

Gegenbauer ($a=1/2$)



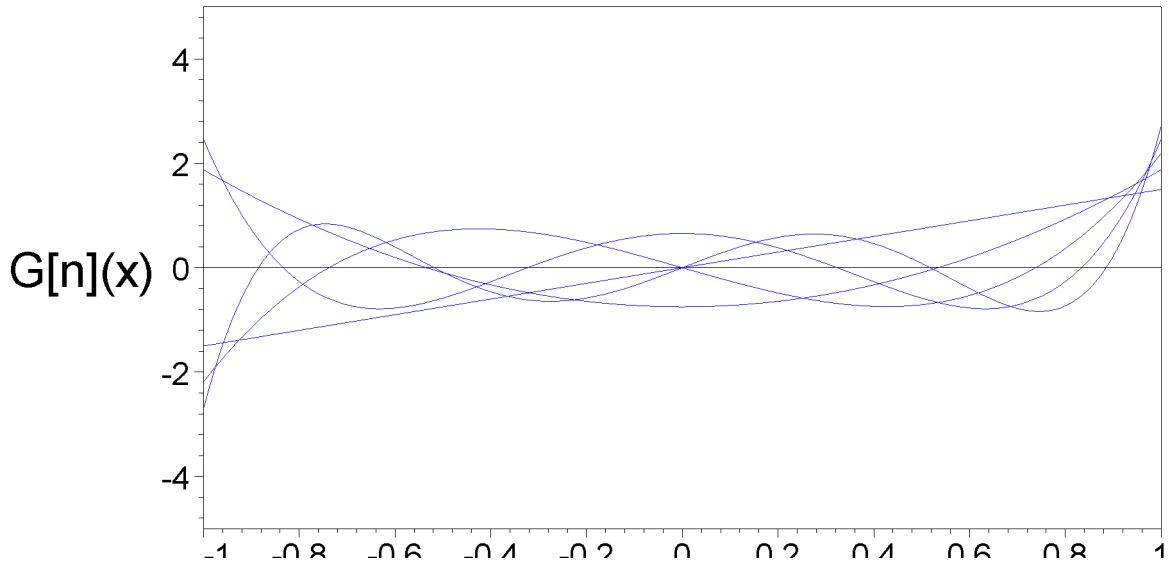
plotpoly $\left([1, 2, 3, 4, 5], G, \frac{5}{8}\right)$

Gegenbauer ($a=5/8$)



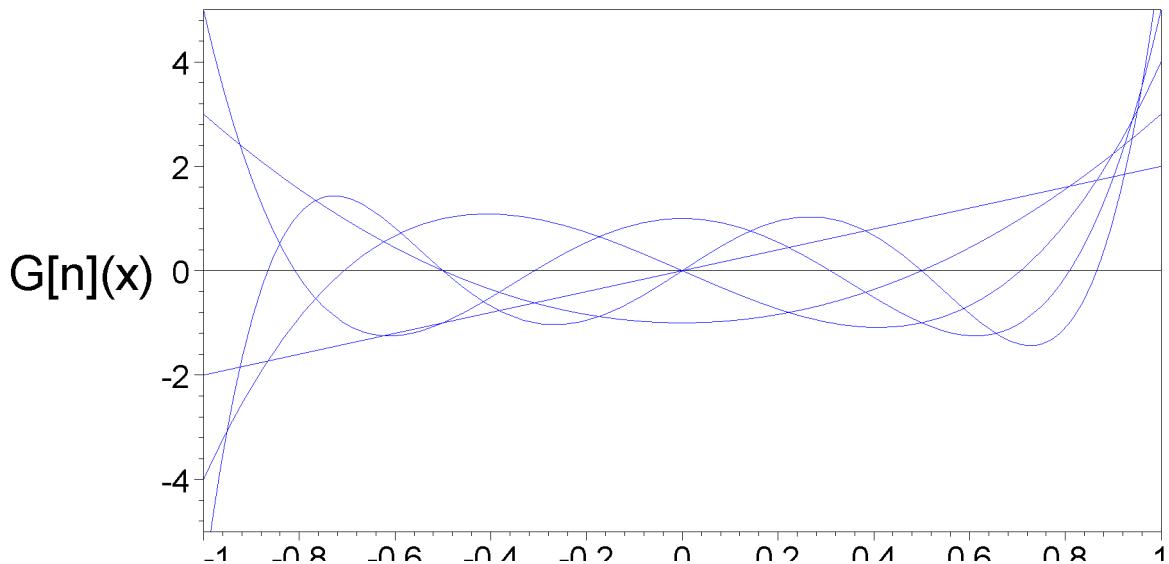
plotpoly $\left([1, 2, 3, 4, 5], G, \frac{3}{4}\right)$

Gegenbauer ($a=3/4$)



```
plotpoly([ 1, 2, 3, 4, 5 ], G, 1)
```

Gegenbauer ($a=1$)



[-] A Numerical Illustration

[Start with some function of t :

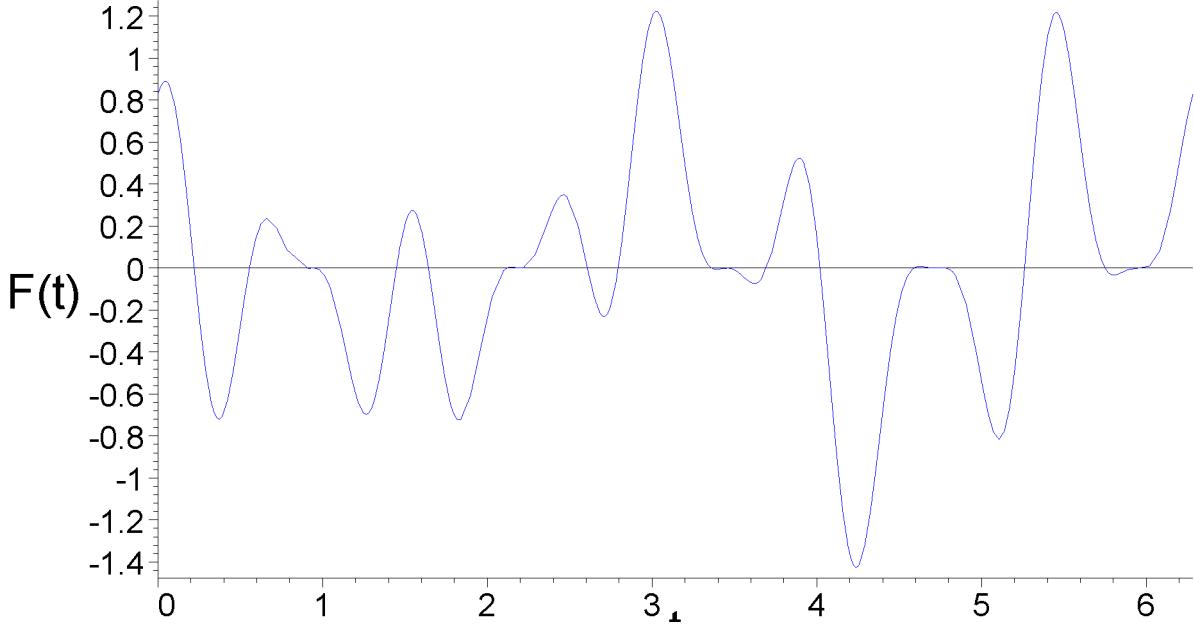
$f := (f1, f2, f3, A1, A2, \phi1, \phi2, t) \rightarrow (1 + \sin(f1 t)) (A1 \cos(f2 t + \phi1) + A2 \cos(f3 t + \phi2))$

$F(t) = f(v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, t + b)$

$$F(t) = (1 + \sin(v_1(t+b))) (A_1 \cos(v_2(t+b) + \phi_1) + A_2 \cos(v_3(t+b) + \phi_2))$$

It looks like this:

```
plot([f(5, 2, 8, .4, .5, .5, .3, t), 0], t = 0 .. 2 π, color = [blue, black], labels = ["t", "F(t)"])
```



To fourth order in t , we have, analytically,

$N := 5$

$$F(t) = \text{series}(f(v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, t+b), t, N)$$

$$F(t) = (1 + \sin(v_1 b)) (A_1 \cos(v_2 b + \phi_1) + A_2 \cos(v_3 b + \phi_2)) + ($$

$$(1 + \sin(v_1 b)) (-A_1 \sin(v_2 b + \phi_1) v_2 - A_2 \sin(v_3 b + \phi_2) v_3)$$

$$+ \cos(v_1 b) v_1 (A_1 \cos(v_2 b + \phi_1) + A_2 \cos(v_3 b + \phi_2))) t + ($$

$$\cos(v_1 b) v_1 (-A_1 \sin(v_2 b + \phi_1) v_2 - A_2 \sin(v_3 b + \phi_2) v_3)$$

$$+ (1 + \sin(v_1 b)) \left(-\frac{1}{2} A_2 \cos(v_3 b + \phi_2) v_3^2 - \frac{1}{2} A_1 \cos(v_2 b + \phi_1) v_2^2 \right)$$

$$- \frac{1}{2} \sin(v_1 b) v_1^2 (A_1 \cos(v_2 b + \phi_1) + A_2 \cos(v_3 b + \phi_2)) \right) t^2 + ($$

$$\cos(v_1 b) v_1 \left(-\frac{1}{2} A_2 \cos(v_3 b + \phi_2) v_3^2 - \frac{1}{2} A_1 \cos(v_2 b + \phi_1) v_2^2 \right)$$

$$- \frac{1}{2} \sin(v_1 b) v_1^2 (-A_1 \sin(v_2 b + \phi_1) v_2 - A_2 \sin(v_3 b + \phi_2) v_3)$$

$$\begin{aligned}
& + (1 + \sin(v_1 b)) \left(\frac{1}{6} A_2 \sin(v_3 b + \phi_2) v_3^3 + \frac{1}{6} A_1 \sin(v_2 b + \phi_1) v_2^3 \right) \\
& - \frac{1}{6} \cos(v_1 b) v_1^3 (A_1 \cos(v_2 b + \phi_1) + A_2 \cos(v_3 b + \phi_2)) \Big) t^3 + \Big(\\
& \frac{1}{24} \sin(v_1 b) v_1^4 (A_1 \cos(v_2 b + \phi_1) + A_2 \cos(v_3 b + \phi_2)) \\
& + \cos(v_1 b) v_1 \left(\frac{1}{6} A_2 \sin(v_3 b + \phi_2) v_3^3 + \frac{1}{6} A_1 \sin(v_2 b + \phi_1) v_2^3 \right) \\
& + (1 + \sin(v_1 b)) \left(\frac{1}{24} A_1 \cos(v_2 b + \phi_1) v_2^4 + \frac{1}{24} A_2 \cos(v_3 b + \phi_2) v_3^4 \right) \\
& - \frac{1}{2} \sin(v_1 b) v_1^2 \left(-\frac{1}{2} A_2 \cos(v_3 b + \phi_2) v_3^2 - \frac{1}{2} A_1 \cos(v_2 b + \phi_1) v_2^2 \right) \\
& - \frac{1}{6} \cos(v_1 b) v_1^3 (-A_1 \sin(v_2 b + \phi_1) v_2 - A_2 \sin(v_3 b + \phi_2) v_3) \Big) t^4 + O(t^5)
\end{aligned}$$

We'll actually use 20th order for numerical calculations.

$S := \text{series}(f(v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, t + b), t, 21)$

Create a function that numerically evaluates the Taylor series representation.

$\text{polyS} := \text{fn}(\text{convert}(S, \text{polynom}), v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, b, t)$

For example,

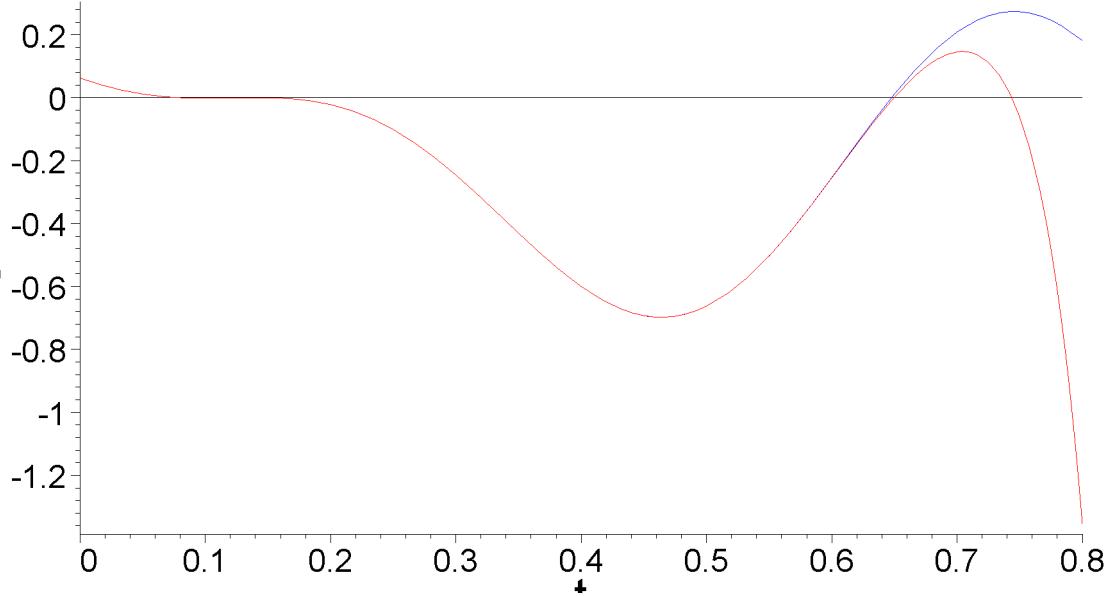
$t0 := .8$

$\text{polyS}(5, 2, 8, .4, .5, .3, t0, t)$

$$\begin{aligned}
& .06207692484 - 1.396001161 t + 6.504389786 t^2 + 32.43092493 t^3 - 210.0827634 t^4 \\
& - 252.8179914 t^5 + 1442.725958 t^6 + 958.6799710 t^7 - 4653.479121 t^8 - 2181.091779 t^9 \\
& + 8953.220365 t^{10} + 3303.351450 t^{11} - 11567.86799 t^{12} - 3556.804969 t^{13} + 10778.73834 t^{14} \\
& + 2855.194048 t^{15} - 7599.977383 t^{16} - 1772.211731 t^{17} + 4199.460941 t^{18} + 875.3937354 t^{19} \\
& - 1868.007524 t^{20}
\end{aligned}$$

Compare the Taylor series with the original function:

$\text{plot}([\text{polyS}(5, 2, 8, .4, .5, .3, t0, t), f(5, 2, 8, .4, .5, .3, t + t0), 0], t = 0 .. .8,$
 $\text{color} = [\text{red}, \text{blue}, \text{black}], \text{labels} = ["t", "F(t)"])$



The Taylor series is good to about $t = .6$.

Check the conversion routines:

```
convert(S, P, t)
simplify(convert(S, polynom) - convert(% , taylor, t))
0
```

Take a look at the first few terms of the Legendre polynomial representation:

```
F(t) = collect(convert(series(f(v1, v2, v3, A1, A2, phi1, phi2, t + b), t, N), P, t),
[seq(Pk(t), k = 1 .. N - 1), A1, A2, sin, cos], factor)
```

$$\begin{aligned} F(t) = & \left(\left(\frac{1}{10} v_2 \left(3 v_1^2 - 10 + v_2^2 \right) \sin(v_2 b + \phi_1) \sin(v_1 b) + \frac{1}{10} v_2 \left(-10 + v_2^2 \right) \sin(v_2 b + \phi_1) \right. \right. \\ & \left. \left. - \frac{1}{10} v_1 \left(-10 + v_1^2 + 3 v_2^2 \right) \cos(v_2 b + \phi_1) \cos(v_1 b) \right) A_1 + \left(\right. \right. \\ & \left. \left. \frac{1}{10} v_3 \left(v_3^2 + 3 v_1^2 - 10 \right) \sin(v_3 b + \phi_2) \sin(v_1 b) + \frac{1}{10} v_3 \left(v_3^2 - 10 \right) \sin(v_3 b + \phi_2) \right. \\ & \left. \left. - \frac{1}{10} v_1 \left(-10 + 3 v_3^2 + v_1^2 \right) \cos(v_3 b + \phi_2) \cos(v_1 b) \right) A_2 \right) P_1(t) + \left(\right. \\ & \left(\frac{1}{42} v_2^4 - \frac{1}{3} v_2^2 + \frac{1}{42} v_1^4 - \frac{1}{3} v_1^2 + \frac{1}{7} v_1^2 v_2^2 \right) \cos(v_2 b + \phi_1) \sin(v_1 b) \\ & \left. + \frac{2}{21} v_1 v_2 \left(v_2^2 - 7 + v_1^2 \right) \cos(v_1 b) \sin(v_2 b + \phi_1) + \frac{1}{42} v_2^2 \left(v_2^2 - 14 \right) \cos(v_2 b + \phi_1) \right) A_1 + \left(\right. \end{aligned}$$

$$\begin{aligned}
& \left(-\frac{1}{3} v_1^2 - \frac{1}{3} v_3^2 + \frac{1}{42} v_3^4 + \frac{1}{42} v_1^4 + \frac{1}{7} v_1^2 v_3^2 \right) \cos(v_3 b + \phi_2) \sin(v_1 b) \\
& + \frac{2}{21} v_1 v_3 \left(v_1^2 + v_3^2 - 7 \right) \cos(v_1 b) \sin(v_3 b + \phi_2) + \frac{1}{42} v_3^2 \left(-14 + v_3^2 \right) \cos(v_3 b + \phi_2) \Big) A_2 \\
P_2(t) & + \left(\left(\frac{1}{15} v_2 \left(3 v_1^2 + v_2^2 \right) \sin(v_2 b + \phi_1) \sin(v_1 b) + \frac{1}{15} \sin(v_2 b + \phi_1) v_2^3 \right. \right. \\
& - \frac{1}{15} v_1 \left(3 v_2^2 + v_1^2 \right) \cos(v_2 b + \phi_1) \cos(v_1 b) \Big) A_1 + \left(\right. \\
& \frac{1}{15} v_3 \left(v_3^2 + 3 v_1^2 \right) \sin(v_3 b + \phi_2) \sin(v_1 b) + \frac{1}{15} \sin(v_3 b + \phi_2) v_3^3 \\
& \left. \left. - \frac{1}{15} v_1 \left(3 v_3^2 + v_1^2 \right) \cos(v_3 b + \phi_2) \cos(v_1 b) \right) A_2 \right) P_3(t) + \left(\right. \\
& \left(\frac{1}{105} v_2^4 + \frac{1}{105} v_1^4 + \frac{2}{35} v_1^2 v_2^2 \right) \cos(v_2 b + \phi_1) \sin(v_1 b) \\
& + \frac{4}{105} v_1 v_2 \left(v_1^2 + v_2^2 \right) \cos(v_1 b) \sin(v_2 b + \phi_1) + \frac{1}{105} \cos(v_2 b + \phi_1) v_2^4 \Big) A_1 + \left(\right. \\
& \left(\frac{1}{105} v_3^4 + \frac{2}{35} v_1^2 v_3^2 + \frac{1}{105} v_1^4 \right) \cos(v_3 b + \phi_2) \sin(v_1 b) \\
& + \frac{4}{105} v_1 v_3 \left(v_3^2 + v_1^2 \right) \cos(v_1 b) \sin(v_3 b + \phi_2) + \frac{1}{105} \cos(v_3 b + \phi_2) v_3^4 \Big) A_2 \Big) P_4(t) + \left(\right. \\
& \left(\frac{1}{20} v_1^2 v_2^2 - \frac{1}{6} v_1^2 + \frac{1}{120} v_2^4 - \frac{1}{6} v_2^2 + \frac{1}{120} v_1^4 + 1 \right) \cos(v_2 b + \phi_1) \sin(v_1 b) \\
& + \frac{1}{30} v_1 v_2 \left(-10 + v_2^2 + v_1^2 \right) \cos(v_1 b) \sin(v_2 b + \phi_1) + \left(-\frac{1}{6} v_2^2 + 1 + \frac{1}{120} v_2^4 \right) \cos(v_2 b + \phi_1) \Big) \\
A_1 & + \left(\left(\frac{1}{20} v_1^2 v_3^2 + \frac{1}{120} v_1^4 + \frac{1}{120} v_3^4 - \frac{1}{6} v_1^2 - \frac{1}{6} v_3^2 + 1 \right) \cos(v_3 b + \phi_2) \sin(v_1 b) \right. \\
& \left. + \frac{1}{30} v_1 v_3 \left(v_3^2 - 10 + v_1^2 \right) \cos(v_1 b) \sin(v_3 b + \phi_2) + \left(-\frac{1}{6} v_3^2 + 1 + \frac{1}{120} v_3^4 \right) \cos(v_3 b + \phi_2) \right) \\
A_2
\end{aligned}$$

Convert the Taylor series to different representations, for example Legendre and Chebyshev:

$$polyP := \text{fn}(\text{convert}(S, P, t), v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, b, t)$$

$$polyT := \text{fn}(\text{convert}(S, T, t), v_1, v_2, v_3, A_1, A_2, \phi_1, \phi_2, b, t)$$

Numerically,

$$\text{polyP}(5, 2, 8, .4, .5, .3, t0, t)$$

$$-4.610469314 P_{14}(t) - 16.51818535 P_{12}(t) - 119.9090086 P_6(t) - .01420962773 P_{20}(t)$$

$$\begin{aligned}
& - 78.42786261 P_8(t) + .1276975151 P_{17}(t) + .01298500508 P_{19}(t) + 67.25551101 P_5(t) \\
& - 40.25374197 P_{10}(t) + 3.713495972 P_{13}(t) + .9220030941 P_{15}(t) - .1415726500 P_{18}(t) \\
& + 27.73912212 P_9(t) - 23.04616294 + 49.84667187 P_7(t) + 12.10422729 P_{11}(t) \\
& - 1.113061850 P_{16}(t) - 100.7218211 P_2(t) - 133.9466093 P_4(t) + 34.47075886 P_1(t) \\
& + 64.53518243 P_3(t)
\end{aligned}$$

polyT(5, 2, 8, .4, .5, .3, t0, t)

$$\begin{aligned}
& -1.554896236 T_{14}(t) - 41.04665867 T_8(t) + 12.99539427 T_9(t) + 50.65780036 T_5(t) \\
& - .3321836094 T_{16}(t) + 1.303804019 T_{13}(t) - 6.177973112 T_{12}(t) + 28.51897285 T_7(t) \\
& + .03640607466 T_{17}(t) - 17.64107694 T_{10}(t) + .003339362086 T_{19}(t) + 73.66882940 T_3(t) \\
& - .003562941600 T_{20}(t) - 78.9192362 T_6(t) - 88.16561101 - 161.2279617 T_2(t) \\
& - 123.5943233 T_4(t) - .03921948765 T_{18}(t) + .2855874466 T_{15}(t) + 4.790239124 T_{11}(t) \\
& + 88.46728399 T_1(t)
\end{aligned}$$

polyS(5, 2, 8, .4, .5, .3, t0, t)

$$\begin{aligned}
& .06207692484 - 1.396001161 t + 6.504389786 t^2 + 32.43092493 t^3 - 210.0827634 t^4 \\
& - 252.8179914 t^5 + 1442.725958 t^6 + 958.6799710 t^7 - 4653.479121 t^8 - 2181.091779 t^9 \\
& + 8953.220365 t^{10} + 3303.351450 t^{11} - 11567.86799 t^{12} - 3556.804969 t^{13} + 10778.73834 t^{14} \\
& + 2855.194048 t^{15} - 7599.977383 t^{16} - 1772.211731 t^{17} + 4199.460941 t^{18} + 875.3937354 t^{19} \\
& - 1868.007524 t^{20}
\end{aligned}$$

The Polynomials Package

Contents of the file c:\Maple\polynomials\polynomials.p

```

=====
# Polynomial conversion functions
=====
# Marc A. Murison
# Astronomical Applications Dept.
# U.S. Naval Observatory
# 3450 Massachusetts Ave., NW
# Washington, DC 20392
# murison@aa.usno.navy.mil
# http://aa.usno.navy.mil/murison/
=====
```

```

#-----
# Convert a Taylor series in x to orthogonal polynomials in x.
#
# Usage: convert( p, ptype, x ), where
#        p      = a polynomial
#        ptype = one of the polynomial types
#        x      = the independent variable (i.e., p = p(x))
#
# Polynomial types are:
#        T    Chebyshev type 1
#        U    Chebyshev type 2
#        P    Legendre
#        H    Hermite
#        L    Laguerre
#-----
polynomials[`convert/T`] := proc( poly::`+`,series , x::name )
    global T, __T;
    T := 'T';
    `polynomials/convert_terms`(poly,x,T,args[3..nargs]);
    subs( __T=T, % );
end:

polynomials[`convert/U`] := proc( poly::`+`,series , x::name )
    global U, __T;
    U := 'U';
    `polynomials/convert_terms`(poly,x,U,args[3..nargs]);
    subs( __T=U, % );
end:

polynomials[`convert/P`] := proc( poly::`+`,series , x::name )
    global P, __T;
    P := 'P';
    `polynomials/convert_terms`(poly,x,P,args[3..nargs]);
    subs( __T=P, % );
end:

polynomials[`convert/H`] := proc( poly::`+`,series , x::name )
    global H, __T;
    H := 'H';
    `polynomials/convert_terms`(poly,x,H,args[3..nargs]);
    subs( __T=H, % );
end:

polynomials[`convert/L`] := proc( poly::`+`,series , x::name )
    global L, __T;
    L := 'L';
    `polynomials/convert_terms`(poly,x,L,args[3..nargs]);
    subs( __T=L, % );
end:

polynomials[`convert/G`] := proc( poly::`+`,series , x::name, a::algebraic )
    global G, __T;
    G := 'G';
    `polynomials/convert_terms`(poly,x,G,a,args[4..nargs]);
    subs( __T=G, % );
end:

```

```

#-----
# Convert an orthogonal polynomial in x to a Taylor series in x.
#
# Usage: convert( p, taylor, x ), where
#        p = a polynomial
#        x = the independent variable (i.e., p = p(x))
#
# Polynomial types are:
#        T    Chebyshev type 1
#        U    Chebyshev type 2
#        P    Legendre
#        H    Hermite
#        L    Laguerre
#-----
polynomials[`convert/taylor`]:= proc( poly::`+`, x::name )
    orthosubs( args );
    collect( %, x );
end:

#-----
# Substitute for T, U, P, H, L, or G, in the polynomial poly,
# the corresponding orthogonal polynomial terms. For example,
#
# > randpoly(x);
#
#
$$x^5 - \frac{47}{21}x^4 - \frac{91}{15}x^3 - \frac{47}{45}x^2 - \frac{61}{35}x + \frac{41}{35}$$

#
# > convert(% ,P,x);
#
#
$$\begin{aligned} & \frac{1222}{21}P[2](x) + \frac{239}{15}P[3](x) - \frac{1618}{45}P[1](x) - \frac{4031}{35}P[4](x) \\ & + \frac{376}{35}P[5](x) \end{aligned}$$

#
# > orthosubs(% ,x);
#
#
$$x^5 - \frac{47}{21}x^4 - \frac{91}{15}x^3 - \frac{47}{45}x^2 - \frac{61}{35}x + \frac{41}{35}$$

#-----
polynomials[orthosubs]:= proc( poly::`+`, x::name )

local N, k, p, pargs, ptype, xpart;

#detect the orthogonal polynomial type Q
xpart := select( has, poly, x );
if type(xpart,`+`) then
    op(1,select(has,poly,x));           #xpart = A*Pn(x) + B*Pm(x) + ...
    op(select(has,[op(%)],x));          #grab a term
    op(select(has,[op(%)],x));          #isolate Q[n](x)
else
    select( has, xpart, x );            #xpart = A*Pn(x)
                                         #isolate Q[n](x)

```

```

fi;
ptype := op([0,0],%);                      #select Q
debug_print(procname,cat("polynomial type = ",ptype),5);

if ptype='G' then
    if nargs=2 then
        pargs := 0,x;
    else
        pargs := args[3],x;
    fi;
else
    pargs := x;
fi;

#determine the polynomial order
N := 0;
for p in poly do
    if has(p,ptype) then
        N := max( N, op([0,1],op(select(has,[op(p)],x))) );
    fi;
od;

#substitute for Q[k](x) the orthogonal polynomial term in x
subs( seq( ptype[k](x)=orthopoly[ptype](k,pargs), k=0..N ), poly );

end:

#-----
# A plotting procedure for demonstration purposes.
#-----
polynomials[plotpoly] := proc( intlist::list(posint), ptype::name )

local p, k, xmin, xmax, ymin, ymax, ytext, ttext, pargs;

if ptype='G' then
    pargs := args[3], x;
else
    pargs := x;
fi;

xmin := -1;
xmax := 1;
ymin := -1;
ymax := 1;
if ptype='L' then
    xmin := 0;
    xmax := 5;
    ymin := -5;
    ymax := 5;
elif ptype='H' then
    xmin := 0;
    xmax := 3;
    ymin := -2;
    ymax := 8;
elif ptype='U' then
    ymin := -5;

```

```

ymax :=      5;
elif ptype='G' and args[3] > 1/2 then
    ymin := -5;
    ymax :=      5;
fi;

p := [];
for k from 1 to nops(intlist) do
    if ptype='H' then
        p := [ op(p),
            plot( orthopoly[ptype](intlist[k],pargs)/intlist[k]^3,
                  x=xmin..xmax, color=blue ) ];
    else
        p := [ op(p),
            plot( orthopoly[ptype](intlist[k],pargs),
                  x=xmin..xmax, color=blue ) ];
    fi;
od;

ytext := cat(ptype,"[n](x)");
if ptype='P' then
    ttext := "Legendre";
elif ptype='T' then
    ttext := "Chebyshev type 1";
elif ptype='U' then
    ttext := "Chebyshev type 2";
elif ptype='H' then
    ttext := "Hermite";
    ytext := cat(ptype,"[n](x)/n^3");
elif ptype='L' then
    ttext := "Laguerre";
elif ptype='G' then
    ttext := cat("Gegenbauer (a=",convert(args[3],string),")");
fi;

plots[display]( [op(p),plot(0,x=xmin..xmax,color=black)],
                view=[xmin..xmax,ymin..ymax],
                title=ttext, labels=["x",ytext] );

end:

#-----
# Initialization function, where we define the workhorse
# routines used by the public polynomials package routines.
#-----
polynomials[init] := proc()

    global `polynomials/convert_terms`, `polynomials/get_horner_coeffs`;

#-----
# The polynomial conversion workhorse routine.
# Not intended for public use.
#-----
`polynomials/convert_terms` := proc( poly::`+`,series}, x::name, ptype::name )

```

```

global __T, `polynomials/get_horner_coeffs`;
local n,          #order of the polynomial
      Q,          #local storage of the polynomial
      r,          #structure to hold the recursion formulae for x*T[n]
      r1,         #structure to hold the recursion formulae for x (usually T[1])
      k, c, p, j, tmp, m, S, a, pargs, astart, i, nterms;

if not type(poly,{polynom(anything,x),series}) then
    ERROR(cat("not a polynomial or series in ",x));
fi;

if type(poly,series) then
    S := convert(poly,polynom);
else
    S := poly;
fi;

__T := '__T';
if ptype='G' then
    if nargs=3 then
        a := 0;
        astart := 4;
    else
        a := args[4];
        astart := 5;
    fi;
    if a=0 then
        ERROR("Computation of G[n](x,0) is erroneous in the polynomials package.");
    fi;
    pargs := __T, a;
else
    astart := 4;
    pargs := __T;
fi;

#assign the recursion formulae for r=x*T[n](x) and r1=x for each polynomial type
r[T]  := (P,n) -> (P[n+1] + P[n-1])/2;
r1[T] := (P)   -> P[1];
r[U]  := (P,n) -> (P[n+1] + P[n-1])/2;
r1[U] := (P)   -> P[1]/2;
r[P]  := (P,n) -> ((n+1)*P[n+1] + n*P[n-1])/(2*n+1);
r1[P] := (P)   -> P[1];
r[H]  := (P,n) -> P[n+1]/2 + n*P[n-1];
r1[H] := (P)   -> P[1]/2;
r[L]  := (P,n) -> (n+1)*(P[n] - P[n+1]) + n*(P[n] - P[n-1]);
r1[L] := (P)   -> 1-P[1];
r[G]  := proc(P,a,n)
        ((n+1)*P[n+1] + (n+2*a-1)*P[n-1])/(2*(n+a));
end;
r1[G] := proc(P,a)
    if a=0 then
        P[1]/2;
    else
        P[1]/(2*a);
    fi;
end;

```

```

#get the horner-form coefficients
c := `polynomials/get_horner_coeffs`(S,x);
n := nops(c) - 1;
debug_print(procname,"horner coefficients",5,c);

#starting with the innermost horner nesting, replace x and T[n]*x with
#equivalents from the recursion formula for the polynomial type T
Q := 0;
for k from 1 to n do
    if k=1 then
        Q := c[n-k+1] + c[n-k+2]*r1[ptype](pargs);
    else
        Q := c[n-k+1] + Q*x;
        Q := collect( Q, [seq(__T[j],j=0..k-1)] );
        if type(Q,`+`) then
            nterms := nops(Q);
        else
            nterms := 1;
        fi;
        tmp := Q;
        for i from 1 to nterms do
            if nterms=1 then
                p := tmp;
            else
                p := op(i,tmp);
            fi;
            if has(p,__T) then
                m := op(1,select(has,p,__T));      #order of T (i.e., T's subscript)
                Q := subs( p=algsubs( __T[m]*x=r1[ptype](pargs,m), p ), Q );
                Q := subs( __T[0]=1,Q);           #T[0]=1 for all orthogonal polynom
            else
                Q := subs( p=subs(x=r1[ptype](pargs),p), Q );
            fi;
        od;
    fi;
od;
Q := collect( Q, [seq(__T[j],j=0..n)], args[astart..nargs] );

#replace T[n] with T[n](x)
for p in Q do
    m := op(1,select(has,p,__T));
    Q := subs( p=subs(__T[m]=__T[m](x),p), Q );
od;

Q;

end:

#-----
# Get the coefficients of the horner form of a polynomial in x.
#
# For example, 1 + 2*x - 3*x^2 becomes 1 + (2 - 3*x)*x, and the
# coefficients are [1,2,-3].
#
# Not intended for public use.
#-----
`polynomials/get_horner_coeffs` := proc( poly::`+`,series }, x::name )

```

```

local S, C, k, n, c, j, m, xpart;

if not type(poly,{polynom(anything,x),series}) then
    ERROR(cat("not a polynomial or series in ",x));
fi;

if type(poly,series) then
    S := convert(poly, polynom);
else
    S := poly;
fi;

S := convert( S, horner, x );
c := [];
n := degree( collect(S,x), x );
for k from 1 to n do
    if type(S,`+`) then                      #S = a + (b + ...)*x
        c := [ op(c), remove(has,S,x) ];
    else                                      #S = (a + ...)*x
        c := [ op(c), 0 ];
    fi;
    xpart := select(has,S,x);
    if xpart=x then
        S := 1;
    else
        if type(xpart, `^` ) or nops(xpart) > 2 then      #xpart=a*b*...*x^m
            m := degree( xpart, x );
        else                                              #xpart=(a + ...)*x^m
            m := degree( op(2,xpart), x );
        fi;
        if m > 1 then
            for j from 2 to m do
                c := [ op(c), 0 ];
                xpart := xpart/x;
                k := k + 1;
            od;
        fi;
        S := xpart/x;
    fi;
od;
[ op(c), S ];

end;

end;

save( polynomials, cat(POLY_PATH,"polynomials.m") );

```